



Universidad Nacional de Loja

Análisis y Diseño de Software

Introducción a GitHub

Martha Suntaxi
Johanna Paz
Victor Jumbo
Andre Montoya
Wilson Iriarte

Ingeniería en Sistemas

Agenda

Introducción a GitHub

Agenda

Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

- Introducción
 - ¿Qué es GitHub?
 - ¿Para que sirve?
 - ¿Qué uso le daremos?
 - ¿Qué herramientas proporciona?
- Instalación
- Comandos Principales
 - Comandos Básicos
 - Trabajos con Ramas
 - Acciones con Archivos
- Conclusiones

Introducción

Introducción a GitHub

Agenda

Introducción

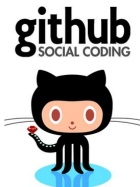
Instalación

Comandos

Conclusiones

Bibliografía

Licencia



¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git.

¿Para que sirve?

- Aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.
- Se Puede contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.

Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.

¿Qué uso le daremos?

En nuestra especialidad “Programación”, fuimos aprendiendo cosas y creando programas de código abierto, fomentando el software libre. Podremos crear una cuenta gratuita y comenzar a subir repositorios de código (o crearlos desde 0), para que con la ayuda de todos ese proyecto mejore; así como también fortalecer los proyectos de los demás para crecer como grupo.

¿Qué herramientas proporciona?

En la actualidad, GitHub es mucho más que un servicio de alojamiento de código. Además de éste, se ofrecen varias herramientas útiles para el trabajo en equipo. Entre ellas, caben destacar:

- Una wiki para el mantenimiento de las distintas versiones de las páginas.
- Un sistema de seguimiento de problemas que permiten a los miembros de tu equipo detallar un problema con tu software o una sugerencia que deseen hacer.
- Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un commit específico.
- Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

Instalación

Introducción a GitHub

Agenda

Introducción

Instalación

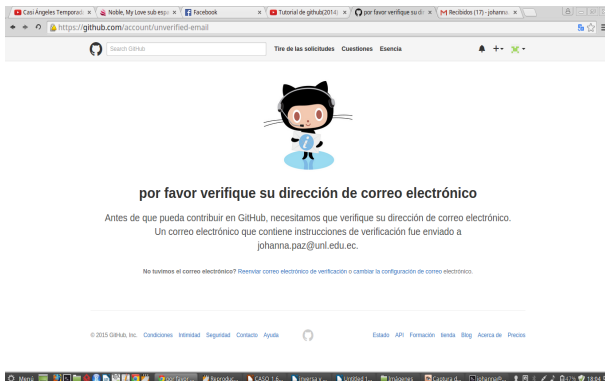
Comandos

Conclusiones

Bibliografía

Licencia

- Crear una cuenta en: <https://github.com/>

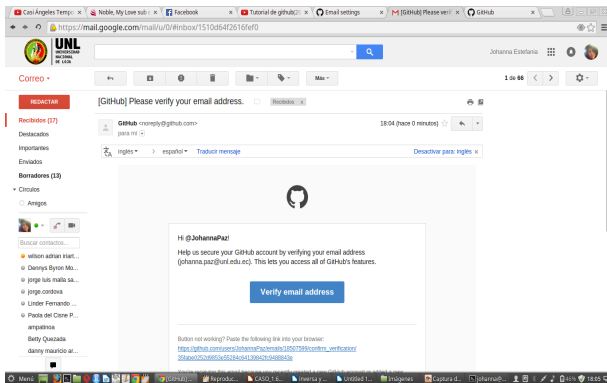


Instalación

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

- Validar la Cuenta en su correo para poder acceder e instalar nuevos repositorios.



Instalación

Introducción a GitHub

Agenda

Introducción

Instalación

Comandos

Conclusiones

Bibliografía

Licencia

- Ahora vamos a instalar Github
- Abrimos una terminal para ejecutar el siguiente comando:

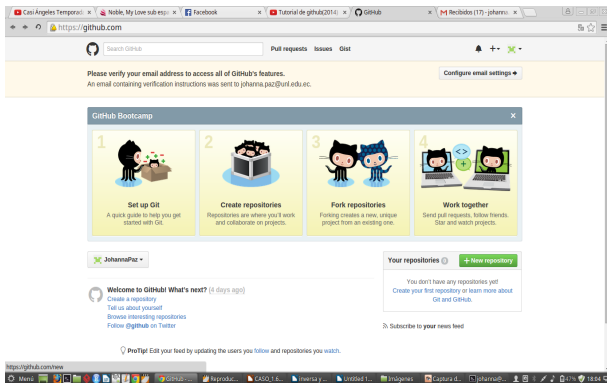
```
sudo apt-get install git-core
```

Instalación

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

- Crear o ingresar en la carpeta que queremos clonar el repositorio

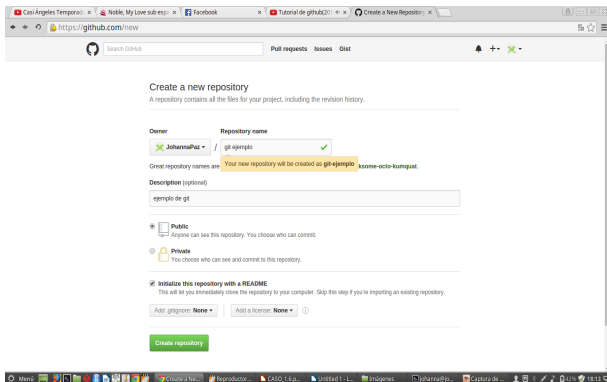


Instalación

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

Ahora vamos a crear el nuevo repositorio así:

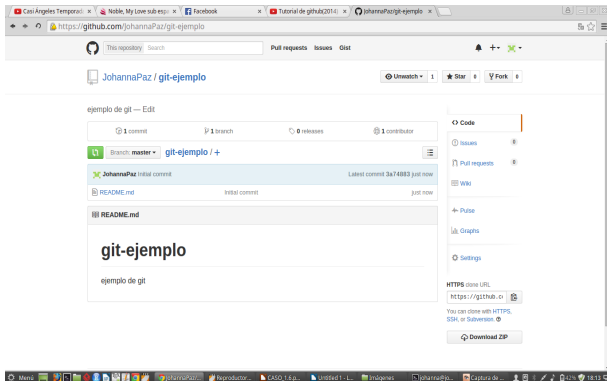


Instalación

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

- Veamos como quedo creado el nuevo repositorio.
git clone
<https://github.com/JohannaPaz/git-ejemplo>

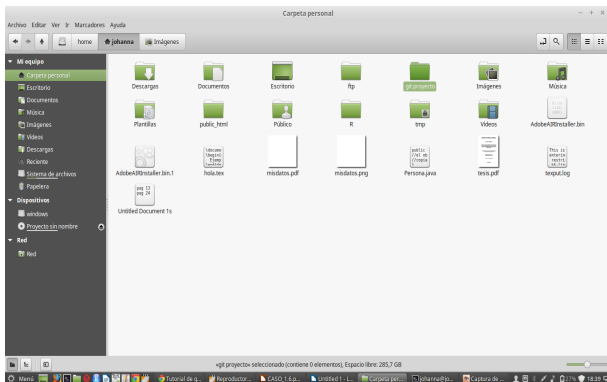


Instalación

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

- Una vez creado nuestro repositorio lo que tenemos que hacer es clonarlo. Para ello tenemos que crear una carpeta en cualquier lugar yo lo he hecho así, la carpeta tiene el nombre de git proyecto



Instalación

Introducción a GitHub

Agenda

Introducción

Instalación

Comandos

Conclusiones

Bibliografía

Licencia

- Ahora accedemos a la consola y escribimos los siguientes comandos para poder clonar la carpeta:

```
cd git proyecto
```

```
ls
```

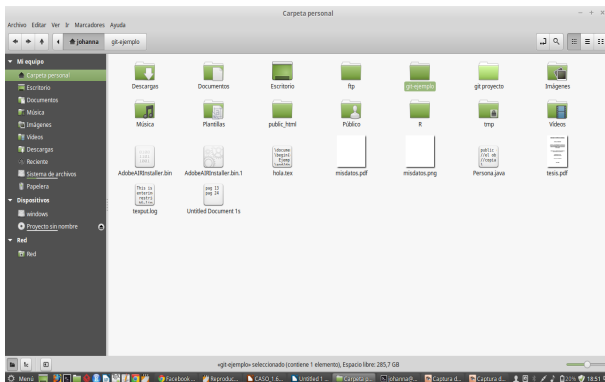
```
git clone https://github.com/JohannaPaz/git-ejemplo.git
```

Instalación

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

- Verificamos que se ha clonado la carpeta así:



Instalación

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

■ Instalar git gui

```
sudo apt-get install gitk giggle git-cola git-gui gitg
```

■ Abrir el gui de git mediante el comando, para eso tenemos q estar en la carpeta donde clonamos el git.

```
git gui
```

■ Poner el nombre del usuario en git

```
git config --global user.name nombre
```

■ Colocar email del usuario

```
git config --global user.email email
```

■ Para asegurarnos de que no exista ningún cambio que nosotros notengamos.

```
git pull origin master
```

■ Subimos los cambios.

```
git push origin master
```


Comandos Básicos

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

■ Inicializar repositorio git

```
git init
```

■ Creación de un fichero README

```
vi README.md
```

■ Añadir a Git los ficheros modificados.

```
git add README.md
```

■ Enviar los cambios hacia GitHub

```
git push origin master
```

Ramas

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

Las ramas son utilizadas para desarrollar funcionalidades aisladas unas de otras.

■ Crear una rama

```
git branch <nombre rama>
```

■ Cambiar de rama

```
git checkout <nombre rama>
```

■ Eliminar una rama

```
git branch -d <nombre rama>
```

■ Renombrar una rama

```
git branch -m <vieja rama> <nueva rama>
```

Acciones con Archivos

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

- **Deshacer cambios locales (reset):** Con este comando descartamos los cambios locales y volvemos al estado que teníamos guardado en el repositorio

```
git reset --hard
```

- Desahacer los cambios realizados en todos los archivos

```
git checkout -- .
```

- Actualizar repositorio local

```
git pull
```

- Fusionar ramas

```
git merge <branch>
```

- Crear una nueva etiqueta en este caso se llama 1.0.0

```
git tag 1.0.0 1b2e1d63ff
```

- Obtener el commit id

```
git log
```

Conclusiones

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

- Es un excelente repositorio para el desarrollo en grupo.
- Podemos obtener conocimientos gracias a que es un repositorio donde se encuentra código profesional de manera gratuita.
- Podemos colaborar con el crecimiento de software libre, subiendo nuestros códigos al repositorio.

Bibliografía

Introducción a GitHub

Agenda
Introducción
Instalación
Comandos
Conclusiones
Bibliografía
Licencia

- 1 L. Castillo, Conociendo GitHub. <https://conociendogithub.readthedocs.org/en/latest/>.
- 2 Y. Torregrosa, Álvaro.Luján Mora, Sergio,2013
<http://rua.ua.es/dspace/handle/10045/26796>.
- 3 Ruiz, José Arístides Valencia.Revista San Gregorio,2015
<http://revista.sangregorio.edu.ec/index.php/RSANG/article/view/66>.

Licencia

Introducción a GitHub

- Agenda
- Introducción
- Instalación
- Comandos
- Conclusiones
- Bibliografía
- Licencia**



Muchas Gracias

